Ricardian contracts

Abstract.

This paper represents a scientific report that concerns *Ricardian Contracts*, a new type of *Smart Contracts* recently introduced to blockchain based technologies. *Smart contracts* were part of the blockchain technology for a while now, but they lacked some crucial aspects of more kinds and that is where *Ricardian Contracts* will play their part. In the following part of this paper, basic aspects of this type of contracts like purpose, flow, validity, versatility and readability will be discussed. It will also bring in some historical context as well as basic terminology in the blockchain field, key differences between the *Smart Contracts* and *Ricardian Contracts*, security aspects when making use of these type of contracts, how they impact the traditional banking system and how is the future shaping up for them.

1. Terminology and historical context

Ricardian Contracts are a type of *Smart Contracts*, so one would need to understand first what are these smart contracts used in blockchain based technologies.

1.1. Blockchain

By a quoted definition, blockchain can be defined as follows:

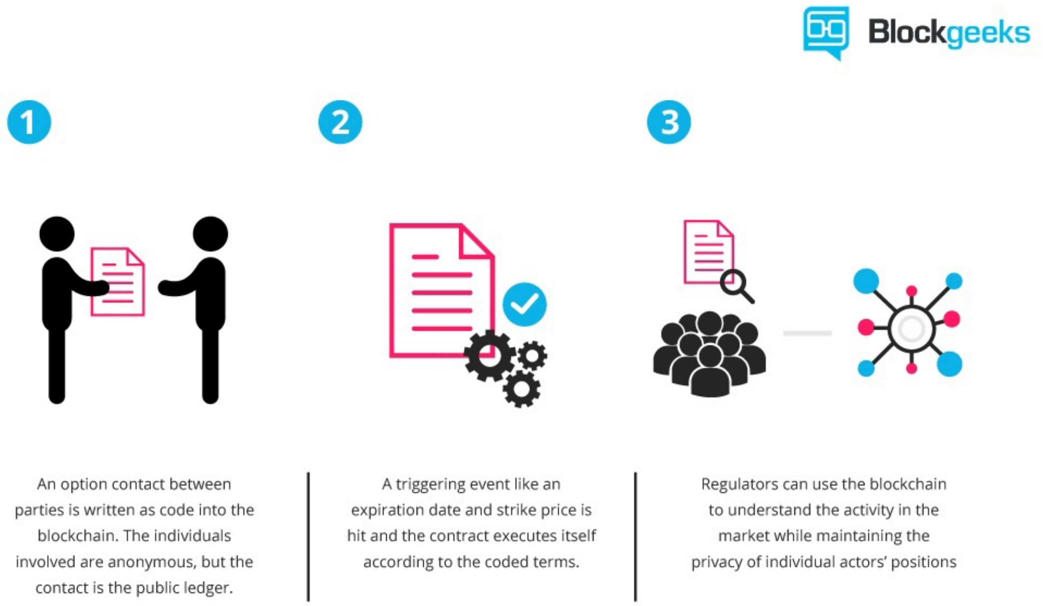
"The blockchain is an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions but virtually everything of value." – Don & Alex Tapscott, authors Blockchain Revolution (2016).

Simply put, blockchain is a timestamped series of immutable records of data, managed by a cluster of computers and not owned by any single entity. Thus, it has no central authority and these blocks of data are secured and bound to each other using cryptographic principles. It also carries **no transaction cost** (only infrastructure costs).

1.2. Smart Contracts

In the blockchain industry, these *Smart Contracts* basically represent contracts that provides the necessary trust during an exchange of money, shares, property and other valuable assets that pertain to an entity, over the internet. These contracts define a set of obligations that are executed through machine code. They are fundamental to processes occurring on the blockchain network where the parties remain anonymous.

In general, the way they work and how they are used in the blockchain network is described below:



An option contact between parties is written as code into the blockchain. The individuals involved are anonymous, but the contact is the public ledger.

A triggering event like an expiration date and strike price is hit and the contract executes itself according to the coded terms.

Regulators can use the blockchain to understand the activity in the market while maintaining the privacy of individual actors' positions

Picture downloaded from:
https://www.elinext.com/wp-content/uploads/2018/02/pic-1-1.jpg

Their main characteristics are that they execute on their own, based on the provided code instructions, they're self-verifying, self-executing (they don't run until someone initiates them), auto-enforcing, immutable – terms cannot be altered, removes the need for third parties and they're cost saving.

The term *Smart Contract* is written using Solidity which is the native language of Ethereum. After the code is written, it is uploaded to the EVM –

Ethereum Virtual Machine that supports executing the code with the help of a universal runtime compiler or browser. Ethereum is not the sole platform where one can create and execute *Smart Contracts*.

There are, however, some drawbacks when it comes to legal aspects, as these *Smart Contracts* agreements don’t have legal bounds. It is why, if anything goes wrong and one of the parties suffers from this, it’s hard to make a lawsuit out of it, let alone make proof of the fraud or scam involved.

The second drawback is that these contracts are not human readable, only computer readable.

1.3. Smart Contracts and IoT

The IoT may be understood like a principle to connect various devices for interoperation. The devices from an IoT network can interact with known and unknown devices. For a better understanding, we can consider a device being a car that meets another while runs towards a specific location. The cars “know” each other and may compute the distance between them, but driving to a new location may find another IoT device like a traffic light. The connection may be established between them, but the smart traffic light is trustworthy ?

Considering an IoT device being a peer of a decentralized peer-to-peer system like blockchain, actions between devices or peers are performed considering the smart contract agreed by them. Using a blockchain based model with a sequential and immutable ledger and consensus, IoT frameworks can take the advantage of performance for automated resources and security by providing:

- distributed system for information sharing
- business terms are embedded for automatization of interactions between peers
- hash-based security
- consensus for intrusion detection and threats mitigation

- low costs. The third party is no more required, because of decentralization of peers and no more need to check the trustworthy of the authority (avoiding DoS attacks)

Smart contracts are neither smart nor contracts. Smart contracts are an alternative to self-execute, partially or fully, self-enforcing or both. It is a distributed app that lives in the blockchain, describes the capabilities of a Thing, the services it offers and how can be accessed.

Whenever a user interacts with a smart contract, all operations are executed by all nodes in blockchain network in a deterministic and reliable way.

Smart contracts can verify the user identity and it's digital signature.

Even if integration of blockchain with the IoT network may look like the perfect solution, there two constraints that must be overcome, for now.

The first is the number of transactions that are supported per second. Currently, Ethereum supports 25 transactions per second(Bitcoin only 7 per second), fact that may be a bottleneck for those IoT networks that host thousands or millions of devices that interact, although the blockchain doesn't limit the number of transactions.

The second is that the IoT network doesn't support complex consensus algorithms. It was designed to support as much as possible connections.

1.4. Ricardian Contracts

Ricardian Contracts are a type of *Smart Contracts* used in blockchain technologies and they're currently drawing a huge interest in the blockchain industry as they patch some of the most important shortcomings of the existing *Smart Contracts*.

Even though it's considered something new for the blockchain, the concept itself dates back in the 1990s, when it was firstly introduced by **Ian Grigg**. *Ricardian Contracts* appeared initially as part of the *Ricardo Payment System* back in 1995 and represents a new type of legal document, considered a pioneer of financial cryptography.

His paper was published in 1998, under the title *Financial Cryptography in 7 Layers*, where he defined *Ricardian Contracts* as follows:

"*A digital contract that defines the terms and conditions of an interaction, between two or more peers, that is cryptographically signed and verified. Importantly it is both human and machine readable and digitally signed*."

Ian Grigg has been working recently as a partner in *Block.One* a leading provider of high-performance blockchain solutions, which is also the same company that adopted *Ricardian Contracts* to the blockchain under the name of *EOS Ricardian Contracts*.

He also warned, in a recent interview, how the traditional banking system is on the verge of collapse and how blockchain technology will influence this outcome in the following years.

1.5. Digital Contracts vs Traditional Contracts

One could always pose a series of questions as to why there is a need for smart contracts in the first place or if there's anything not right in using the traditional contracts.

For a quick answer, traditional contracts are not wrong in any way, except for a flaw deriving right from their format: being written in a human language, they are prone to interpretations.

A human language can create ambiguity that allows lawyers to pave escape routes from the contract clauses.

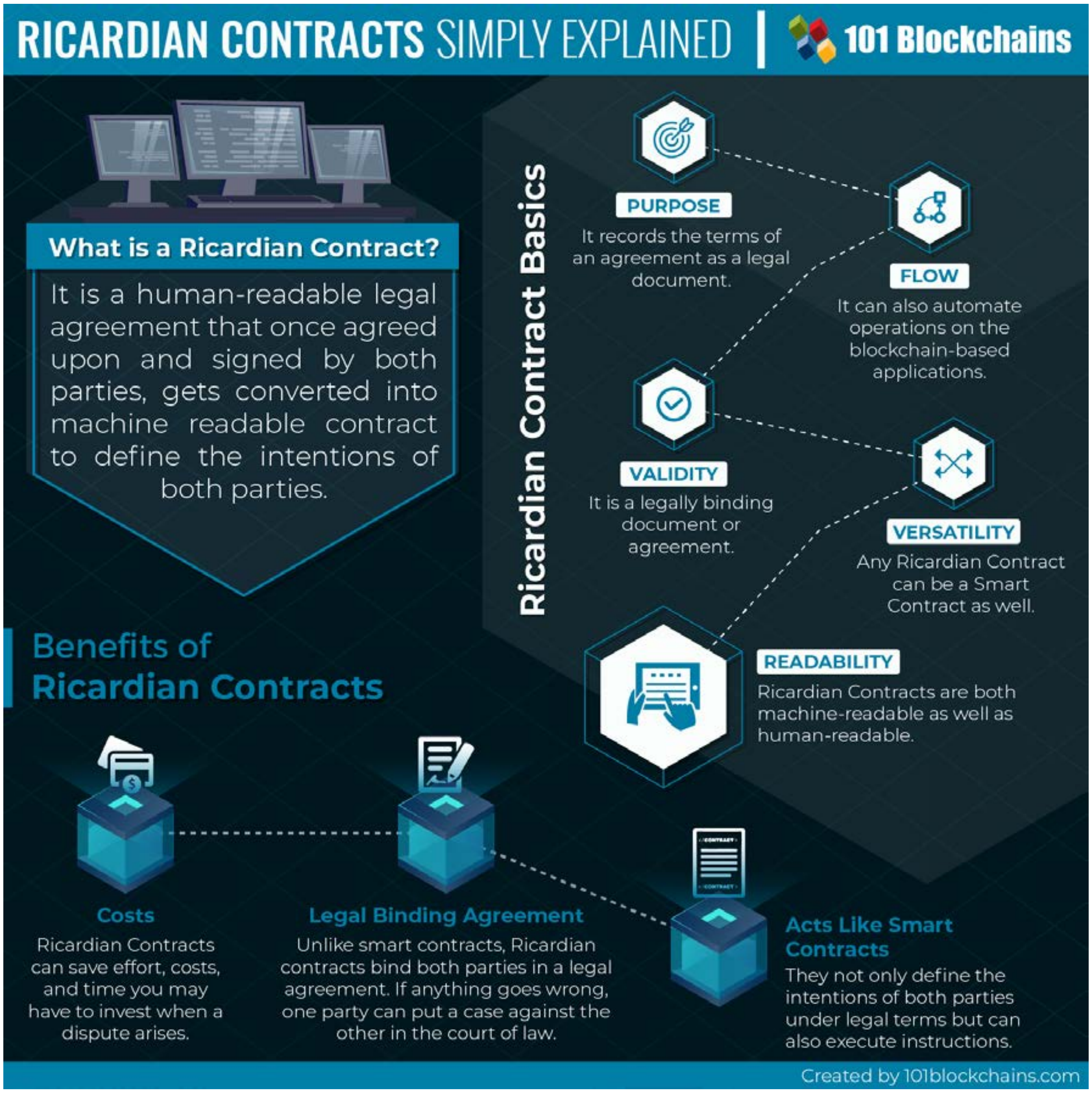
Digital, smart contracts leave no such openings for them and the same is for the involved parties, that won't go for different interpretations when forming a contract.

It is why, as described before about *Ricardian Contracts*, relying on the public judicial system, for dealing with potential lawsuits, is not a concern anymore; these situations were costly and time-consuming.

2. Ricardian Contracts Basics

As mentioned before, *Ricardian Contracts* represent a sort of digital document that acts as an agreement between parties, defining terms and conditions for interaction between these.

The *Ricardian Contract* basics are covered below:



RICARDIAN CONTRACTS SIMPLY EXPLAINED | 101 Blockchains

What is a Ricardian Contract?

It is a human-readable legal agreement that once agreed upon and signed by both parties, gets converted into machine readable contract to define the intentions of both parties.

Ricardian Contract Basics

PURPOSE

It records the terms of an agreement as a legal document.

FLOW

It can also automate operations on the blockchain-based applications.

VALIDITY

It is a legally binding document or agreement.

VERSATILITY

Any Ricardian Contract can be a Smart Contract as well.

READABILITY

Ricardian Contracts are both machine-readable as well as human-readable.

Benefits of Ricardian Contracts

Costs

Ricardian Contracts can save effort, costs, and time you may have to invest when a dispute arises.

Legal Binding Agreement

Unlike smart contracts, Ricardian contracts bind both parties in a legal agreement. If anything goes wrong, one party can put a case against the other in the court of law.

Acts Like Smart Contracts

They not only define the intentions of both parties under legal terms but can also execute instructions.

Created by 101blockchains.com

Picture downloaded from:

https://101blockchains.com/wp-content/uploads/2018/10/Ricardian_Contracts.png

Moreover, compared to *Smart Contracts*, they are cryptographically signed and verified. Also, Ricardian Contracts cope for the shortcomings of *Smart Contracts*, because they are a legal binding contract and are also available in a human-readable text format that is easy to understand even for ordinary people (not only lawyers). This legal binding happens before the execution of the actions on the blockchain network, described in the contract.

Ricardian Contracts are versatile as they can act as a *Smart Contract* as well – because they are *Smart Contracts* at their core.

Because of these basic aspects of *Ricardian Contracts*, one can easily identify the benefits that come by implementing this type of contract in the blockchain: they bind the implied parties in a legal agreement, where if anything goes wrong, one party can bring a lawsuit against another in court, it can save effort, costs and time that may have to be invested in case of these potential lawsuits can act as smart contracts as well, not only defining intentions of the implied parties, but also executing instructions as well.

The before mentioned aspects of *Ricardian Contracts* constitute some of the key differences between them and the former *Smart Contracts*. Other differences would be that, while *Smart Contracts* automate actions on a blockchain application, they do have their limitations in this context, as one cannot have a clear idea of what happens in many scenarios. In this case, this type of contracts can't be used to automate something that is not sure to happen. It can be said that *Smart Contracts* lack the ability to evolve in such scenarios provided the absence of a legal framework, present under *Ricardian Contracts*.

This legal framework added in the *Ricardian Contracts* brings clarity to the intentions and actions defined in the terms and conditions. It describes the parties involved, their representatives (if any), the scope of the contract and what are the applicable consequences (if any) regarding any action.

In addition, the concept introduced by Ian Grigg can be used to add authenticity for processes which involve buying or selling an asset over the internet or the blockchain network, defining, in legal terms, what is the item that is bought or sold, under what legal term, the implied participants and any additional (legal) information about the whole exchange.

Summarizing, *Ricardian Contracts* contain the following information: parties, as in how many are involved, who is making the agreement and by whom they're represented. The validity of the contract in time is another piece of information that can reside in this type of contract. More precisely, the time window during which the terms and conditions are valid can be determined or not (lifetime contract). As an example, the terms defined under a certain contract must be met within half a year, otherwise the contract and everything it supposes is void.

Exceptions can also be specified for some rare situations: for example, actions to be taken if one of the parties cease to exist. The number of such conditions (exceptional cases or not) can be added as needed, without limitation.

2.1. Ricardian Contracts vs Smart Contracts

In the following, there is a simple comparison table illustrating the key differences between *Smart Contracts* and *Ricardian Contracts*.

	Smart Contract	**Ricardian Contract**
Purpose	• execute the agreement of terms and conditions	• record the terms and conditions as a legal document
Flow	• automatization of actions for blockchain applications	• automate also operations on this type of applications
Validity	• not a legal bind for the parties involved	• legal bind for the involved parties
Versatility	• cannot substitute Ricardian Contracts	• can act as a smart contract
Readability	• machine readable, not (necessarily) human readable	• both machine readable and human readable

The way these human readable legal contracts work is well established. Although in more complex cases lawyers may be needed, it is only to create the actual agreement that will facilitate the read and understanding part for the implied parties, after which they can agree upon and sign. Afterwards, it can be hashed so it can be used by software to run on the blockchain platform.

Ricardian Contracts are also **live contracts**, meaning that they can alter after the execution of an event. More precisely, taking the example of a car being sold by a party and bought by the other, an important clause could be constituted by the need to contact some authority that can confirm if the party implied as the seller is the actual owner of the vehicle; with that information taken into account in the contract, a new version of the contract appears. This is the manner in which *Ricardian Contracts* execute different actions and, with the help of these, move forward to reach a logical conclusion, based on the outcome of each event.

2.2. Ricardian Contracts – Security Aspects

Once such contract is prepared, it is digitally signed, hashed and the referencing to this contract will be done using the obtained hash value.

For security reasons, an extra hidden signature is used. The actual agreement (the signing of the contract) is done using private keys, after which its hash is used to attach the hidden signature to the contract.

In general, *Ricardian Contracts* are said to be of high security, using cryptographic signatures. For example, their unique hash value identifier allows a tampered-free existence, after the contract is turned into the machine-readable form. Moreover, this technique allows protection against a common used method when it comes to legal agreements, called boiling frog, where an issuer keeps changing the terms of the agreement during the execution.

The usage of private keys as a security measure also has side benefits: it allows to track the parties involved and hold any of them accountable if any deviations from the agreement would occur.

In general, there’s a small misconception that it is correct to equate *Smart Contracts* with *Ricardian Contracts*. Despite sharing a number of similarities, the two terms are independent one of another. It is right to assume that one could implement a *Ricardian Contract* as a *Smart Contract*, but it's not necessarily true that every *Ricardian Contract* is a *Smart Contract* just the same. Also, the same goes the other way around: not any *Smart Contract* is a *Ricardian Contract*.

Smart Contracts refer to a type of digital agreement where parties have already agreed upon and that executes automatically, without getting modified along the way, whereas a *Ricardian Contract* follows a model that records the *intentions* and *actions*, no matter if the execution took place or not.

It is said that in the near future, more interaction will occur between this new types of contracts that Ian Grigg proposed initially and the former *Smart Contracts* and transactions would probably rely on the basis of different hybrid forms.

2.3. Ricardian Contracts – concerns and limitations

Ricardian Contracts are still in their prime, which means they still lack some finishing touches. A question left still unanswered is who is enforcing the contract itself. Also, how arbitration can be integrated into the EOS ecosystem or how can users bring up claims or complaints. It's possible however, the very least, to have the contract let the arbiters decide about original intent as well as obligations.

2.4. Ricardian Contracts – current use and applications

One of the first applications to make use of the *Ricardian Contracts* is OpenBazaar, an online marketplace where one can buy and / or sell anything. This type of contract is used during any exchange of goods or assets between two parties to track liability. It also tracks the legitimacy of the legal contract that parties agree upon and sign before proceeding further.

This is one of the main reasons *Ricardian Contracts* can have a huge impact and major use in the e-commerce industry, adding that extra layer of security for its users. More thorough details of the use of these contracts and the actual implementation on OpenBazaar can be found on their blog.

2.5. Ricardian Contracts – the future

Ricardian Contracts can have a larger spectrum of use compared to the former *Smart Contracts*. Their main use still resides under financial transactions on the blockchain, but this is not a limitation as this new type of contract formalize responsibilities as a series of legal terms. It is the reason for which most experts say the use of these contracts will go mainstream in the near future on EOS.

References:

1. https://medium.com/contract-vault/ricardian-contracts-101-3faa703022cc
2. https://steemit.com/eos/@iang/towards-a-ricardian-constitution
3. https://medium.com/humanizing-the-singularity/ian-grigg-on-how-the-banking-system-is-about-to-collapse-and-how-to-fix-it-c6c8c1bb6681
4. https://medium.com/ltonetwork/ricardian-contracts-legally-binding-agreements-on-the-blockchain-4c103f120707
5. https://www.packtpub.com/mapt/book/big_data_and_business_intelligence/9781787125445/6/ch06lvl1sec46/ricardian-contracts
6. http://iang.org/papers/fc7.html
7. https://www.openbazaar.org/
8. http://iang.org/ricardian/
9. https://en.wikipedia.org/wiki/Boiling_frog
10. https://www.elinext.com/industries/financial/trends/smart-vs-ricardian-contracts/
11. https://medium.com/datadriveninvestor/ricardian-contracts-can-become-the-next-generation-of-smart-contracts-b77b605d9dda
12. https://medium.com/ltonetwork/ricardian-contracts-legally-binding-agreements-on-the-blockchain-4c103f120707
13. https://datafloq.com/read/ricardian-contracts-killer-application-blockchain/6035
14. https://coincentral.com/ricardian-smart-contracts/
15. https://medium.com/smartz-blog/how-blockchain-and-smart-contracts-can-impact-iot-f9e77ebe02ab
16. https://blockgeeks.com/guides/blockchain-consensus/
17. https://arxiv.org/pdf/1901.10582.pdf